Algebraic Multilevel Methods for Edge Elements

Ronan Perrussel, Laurent Nicolas, François Musy, Laurent Krähenbühl, Michelle Schatzman, Clair Poignard

Algebraic Multilevel Methods for Edge Elements

R. Perrussel[1], L. Nicolas[2], F. Musy[3], L. Krähenbühl[2], M. Schatzman[3], and C. Poignard[3]

[1]Institut Camille Jordan (ICJ) and CEGELY, Ecole Centrale de Lyon 69134 Ecully Cedex, France
[2]CEGELY, Ecole Centrale de Lyon 69134 Ecully Cedex, France
[3]ICJ, Ecole Centrale de Lyon 69134 Ecully Cedex, France

An algebraic multilevel method is proposed for the resolution of linear systems coming from an edge-element discretization of EM models. Graph-flow problems are introduced to ensure a natural compatibility condition linking nodal and edge interlevel transfer operators. The efficiency of our method is compared to classical solvers on two-dimensional and three-dimensional eddy current problems.

***Index Terms*—Algebraic multigrid methods, edge elements.**

I. Introduction

THE finite-element discretization of partial differential equations (PDEs) leads to large sparse linear systems. This is often the most time-consuming part of the finite-element computations. This part can be optimized in view of specific applications. The *multilevel approach*, also called multigrid, consists of considering the linear system at different mesh scales, which can substantially reduce the computational time. Although the multilevel concept is relatively generic, the practical components are narrowly linked to PDE, finite-element, and mesh properties. In this paper, we deal with EM models discretized by the lowest-order edge elements on an unstructured mesh, i.e., without a hierarchy of nested grids.

The starting point is the construction of coarse-nodal and coarse-edge functions which satisfy a natural compatibility relation: The gradient of coarse-nodal functions are linear combinations of coarse-edge functions. This relation is introduced as a constraint in an energy-minimization problem for constructing coarse bases. By linking the compatibility relation to graph-flow problems, the minimization problem can be reduced to a linear system. The efficiency of our approach is compared to classical solvers on two-dimensional (2-D) and three-dimensional (3-D) eddy current problems.

II. Formulation

The following problem has to be solved on a domain Ω

$$\begin{cases} \text{Find } \mathbf{E} \in V_h \text{ such that : } \ a(\mathbf{E}, \mathbf{E}') = F(\mathbf{E}'), \forall \mathbf{E}' \in V_0 \\ \text{with } a(\mathbf{E}, \mathbf{E}') = \int_\Omega \delta \ \text{curl}\ \mathbf{E} \cdot \text{curl}\ \mathbf{E}' + \int_\Omega \alpha \mathbf{E} \cdot \mathbf{E}'. \end{cases} \tag{1}$$

V_0 is an edge element subspace of $\mathbf{H}(\text{curl}, \Omega)$, V_h is the affine space taking into account essential boundary conditions, F is a source term, and δ and α are strictly positive functions. This formulation includes many static and transient EM models. It leads to solve the linear system

$$Ax = b. \tag{2}$$

where the components of x are the coefficients of the solution in the edge-element basis.

III. Algebraic Multilevel Method

A. A Simple Two-Level Case

Suppose that two levels of discretization are known for the problem under consideration. For instance, one could have discretizations on two nested meshes T_H and T_h where the subscript h is used for the *fine* level and H for the *coarse* level. Thus, A_h, x_h, and b_h are, respectively, the matrix, the solution vector and the right-hand side on the fine level. A *prolongation* matrix P transfers information from the coarse to the fine level. The transposed matrix P^T is called *restriction*.

Two complementary steps are needed for the two-level algorithm [1, Section 1.5].

1) *Smoothing*. The "oscillating" part of the error is damped by a linear iteration called smoother S (Gauss–Seidel type methods are often used) $x_h \leftarrow S(A_h, x_h, b_h)$. The new residual $r_h = A_h x_h - b_h$ is then transferred to the coarse level: $r_H \leftarrow P^T r_h$.
2) *Correction*. The "smooth" part of the error is computed on the coarse level $\theta_H \leftarrow A_H^{-1} r_H$ and is prolongated to the fine level as a correction: $x_h \leftarrow x_h - P\theta_H$.

The method is iterative and the two steps are repeated until the norm of the residual $\|r_h\|$ is sufficiently small. More complex variants are possible, using for instance pre- and postsmoothing. Replacing the correction step by a two-level method leads to the recursively defined multilevel method.

B. Algebraic Multilevel

Using a hierarchy of nested grids is the straightest way to implement multilevel techniques. Nevertheless, in some applications, only information at the fine level is available for building coarse levels; this is the case for unstructured meshes. Algebraic strategies must then be followed.

The main task is to define a coarse basis or equivalently the prolongation matrix P. For the recursive application of the multilevel method, the coarse matrix is assembled by the Galerkin product $A_H = P^T A_h P$.

C. Edge Element Features

Hiptmair [2] and Arnold *et al.* [3] have proposed appropriate smoothers for edge elements, specifically dealing with the kernel of the curl operator. For algebraic multilevel methods,

Digital Object Identifier 10.1109/TMAG.2006.871604

the construction of the prolongation matrix must also be considered. Due to relations between first-order conforming nodal elements and lowest-order edge elements [4], an important compatibility relation must be ensured

$$\text{grad } \phi_n^H = \sum_{e=1}^{E^H} G_{\text{en}}^H \lambda_e^H, \quad \forall n \in \{1, \ldots, N^H\} \tag{3}$$

where $(\phi_n^H)_{n=1\ldots N^H}$ is the coarse-nodal basis, $(\lambda_e^H)_{e=1\ldots E^H}$ is the coarse-edge basis, and G^H is the discrete analog of the gradient operator on the coarse level. The matrix G^H can be viewed as the node-edge incidence matrix of an oriented coarse graph $\mathcal{S}^H$ defined by the relations

$$\begin{array}{l} e \text{ index of} \\ \text{an edge in } \mathcal{S}^H \end{array} \Leftrightarrow G_{\text{en}}^H = \begin{cases} -1, & \text{if node } n \text{ is the origin} \\ +1, & \text{if node } n \text{ is the end} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Enforcing (3) is a main issue for algebraic multilevel methods, as it was first highlighted by Reitzinger and Schöberl [5]. The compatibility condition was also used by Bochev *et al.* in [6], [7].

Edge and nodal coarse bases are constructed so as to satisfy the inclusion of finite-element spaces, the "coarse" being included in the "fine," which is expressed by the following algebraic relations:

$$\begin{aligned} \phi_n^H &= \sum_{p=1}^{N^h} P_{pn}^{\text{nod}} \phi_p^h, \quad \forall n \in \{1, \ldots, N^H\} \\ \lambda_e^H &= \sum_{i=1}^{E^h} P_{\text{ie}}^{\text{edg}} \lambda_i^h, \quad \forall e \in \{1, \ldots, E^H\}. \end{aligned} \tag{5}$$

The matrices P^{nod} and P^{edg} are, respectively, the nodal and edge prolongation matrices which have to be constructed.

The analog of relation (3) is also assumed to be satisfied at the fine level

$$\text{grad } \phi_p^h = \sum_{i=1}^{E^h} G_{ip}^h \lambda_i^h, \quad \forall p \in \{1, \ldots, N^h\}. \tag{6}$$

From relations (5) and (6), the compatibility condition (3) is algebraically written

$$P^{\text{edg}} G^H = G^h P^{\text{nod}}. \tag{7}$$

IV. New Approach

In order to compute an efficient coarse basis verifying the compatibility relation, the "energy-minimizing coarse basis" concept from [8] is applied. First, a decomposition of the domain into overlapping subdomains is introduced $\Omega = \cup_{n=1}^{N^H} \Omega_n$. The support of the coarse nodal function ϕ_n^H is enforced to be included in Ω_n, by setting equal to zero values in the nth column of P^{nod}.

Moreover, in order to ensure that the constant functions belong to the coarse nodal space, the sum of each row of P^{nod} is enforced to be equal to one.

We compute the prolongation matrix P^{nod} by usual techniques, for example, by smoothed aggregation [9].

Then, we define a node-edge incidence matrix G^H or, equivalently, an oriented coarse graph such that for any coarse edge e, whose extremities are nodes n and m, Ω_n and Ω_m intersect. Thus, for each coarse edge e, we introduce the subdomain $\mathcal{U}_e = \Omega_n \cap \Omega_m$ and the coarse edge function λ_e^H whose support is enforced to be in $\overline{\mathcal{U}_e}$, by setting equal to zero values in the eth row of P^{edg}.

Then, we solve the following optimization problem:

$$\begin{cases} \text{To find } P^{\text{edg}} \text{ minimizing } \sum_{e=1}^{E^H} \beta_e^T K_e \beta_e \\ \text{under the constraint (7).} \end{cases} \tag{8}$$

Here, β_e is the vector where the nonzero values of the eth column of P^{edg} are gathered, and K_e is a symmetric positive definite matrix. More precisely, K_e may be defined by the discretization of the form a from (1) on $\mathcal{U}_e$, but variants can also be introduced to minimize the overhead of computing the solution of (8).

V. Solving the Optimization Problem

A. Main Concept

For each fine egde index i such that a coarse edge of index e exists with the coefficient P_{ie}^{edg} not enforced to vanish, we introduce a matrix $G^{\text{H,i}}$. This matrix is the node-edge incidence matrix of the subgraph $\mathcal{S}^{\text{H,i}}$ obtained from the coarse graph $\mathcal{S}^H$ by removing the edges of index e for which the coefficient $P_{\text{ie}}^{\text{edg}}$ is enforced to vanish. Let us denote that P_i^{edg} is the row vector defined from the components of the ith row of P^{edg} by extracting the components with edge indices in $\mathcal{S}^{\text{H,i}}$, and $(G^h P^{\text{nod}})_i$ is the row vector defined from the components of the ith row of $(G^h P^{\text{nod}})$ by extracting the components with node indices in $\mathcal{S}^{\text{H,i}}$.

Then, it is proved in [10] that a necessary and sufficient condition of existence of a solution P^{edg} to (7), for any P^{nod}, is the connectivity of subgraphs $\mathcal{S}^{\text{H,i}}$. It is also shown that to find a solution P^{edg} satisfying (7) is equivalent to solve the flow problems

$$P_i^{\text{edg}} G^{\text{H,i}} = (G^h P^{\text{nod}})_i. \tag{9}$$

The solution of such a linear system can be written as

$$P_i^{\text{edg}} = \left(P_i^{\text{edg}}\right)' + \left(P_i^{\text{edg}}\right)'' \tag{10}$$

where in a graph context

- the term $(P_i^{\text{edg}})'$ is a particular solution of the flow problem, which can be computed from a spanning tree of the subgraph $\mathcal{S}^{\text{H,i}}$;
- the transposed of $(P_i^{\text{edg}})''$ belongs to the kernel of $(G^{\text{H,i}})^T$, which can be defined from a set of independent cycles of the subgraph $\mathcal{S}^{\text{H,i}}$.

Thus, the degrees of freedom for the minimization phase are the components in the kernel of $(G^{\text{H,i}})^T$ for each i, for which we can easily build bases.

We also observe that, whatever the matrix P^{nod} is, a matrix P^{edg} satisfying (7) can be built from the rows $(P_i^{\text{edg}})'$; it can be

used as prolongation matrix, but the energy-minimization property is not satisfied.

B. From the Optimization Problem to a Linear System

Introducing appropriate numbering and projection operators for the support constraints, problem (8) is reduced to the resolution of the linear system

$$(B^T DB)\Gamma = -B^T D\tilde{P}^{\mathrm{edg}} \tag{11}$$

where we have gathered information from all the subgraphs.

- The solution Γ is the vector whose components give the coefficients of $((P_i^{\mathrm{edg}})'')^T$ in the bases of the kernel of $(G^{\mathrm{H,i}})^T$.
- The matrix B gathers the basis vectors of these different kernels. It is a sparse full-rank matrix which is assembled during the resolution of flow problems (9).
- The matrix D is block diagonal and its diagonal blocks are the matrices K_e involved in (8).
- The vector $\tilde{P}^{\mathrm{edg}}$ gathers the particular solutions $(P_i^{\mathrm{edg}})'$ from all flow problems (9).

C. Properties of System (11)

The matrices K_e being symmetric positive definite (SPD), the matrix $B^T DB$ is SPD, and we can use the conjugate gradient (CG) to solve the system. In most cases, the matrix is not assembled; we have only to compute matrix-vector products, i.e., operations with B, D, and B^T.

For evaluating the behavior of the CG method on system (11), a rough estimate of the conditioning number of the matrix is given by the inequality

$$\mathrm{cond}_2(B^T DB) \leq \mathrm{cond}_2(B^T B)\mathrm{cond}_2(D) \tag{12}$$

where cond denotes the conditioning number relative to the 2-norm.

The matrix $B^T B$ is similar to a block-diagonal matrix whose blocks are of the form $(\tilde{B}^{\mathrm{H,i}})^T \tilde{B}^{\mathrm{H,i}}$ where $\tilde{B}^{\mathrm{H,i}}$ gathers the basis of the kernel of $(G^{\mathrm{H,i}})^T$. Therefore, the conditioning of $B^T B$ remains low and independent of the global dimension of the problem.

Depending on the choice of the matrices K_e, the conditioning of D may be slightly dependent of the global dimension.

Anyway, observe that the system (11) has not to be solved accurately because its solution is only needed to improve the convergence of iterative methods for the initial system (2).

VI. Numerical Examples and Comments

A. Description of Test Problems

A 2-D eddy current problem in a L-shape conducting domain Ω is solved (Fig. 1). An $\mathbf{E}$-field formulation is used where an exterior magnetic field $\mathbf{H}$ is imposed on the domain boundary. An implicit Euler scheme with time parameter $\triangle t$ is used for time discretization. A problem similar to (1) with the coefficients $\delta = \triangle t/\mu$ and $\alpha = \sigma$ has to be solved at each time step. The source term is given by the electric field $\mathbf{E}$ at the previous instant and $\partial \mathbf{H}_z/\partial t$ on the exterior boundary at the current instant.

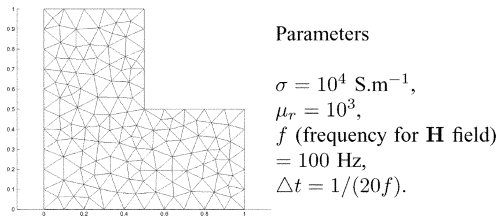

Fig. 1. Domain Ω and parameters of the problem.

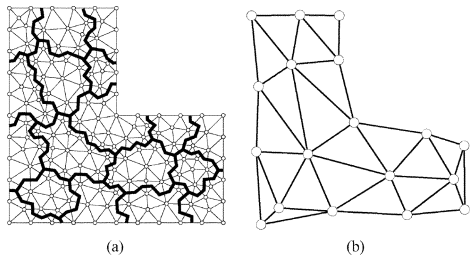

Fig. 2. Fine graph with partition in bold lines and coarse graph. (a) Initial graph and partition in bold lines. (b) Representation of the coarse graph related to 2(a).

In order to evaluate the efficiency according to the increase of the number of degrees of freedom (DOFs), different meshes are used with a decreasing maximal diameter $h_{\max}$ of the mesh elements.

As Reitzinger and Schöberl do in [5], we introduce a partition of the fine nodes:

$$\{1, \ldots, N^h\} = \bigcup_{n=1}^{N^H} H_n. \tag{13}$$

Fig. 2(a) illustrates an automatically determined partition where node aggregates are separated by bold lines. This partition induces a decomposition of domain Ω into overlapping subdomains $\tilde{\Omega}_n$ defined by

$$\tilde{\Omega}_n = \bigcup_{p \in H_n} \mathrm{supp}\left(\phi_p^h\right). \tag{14}$$

The graph $\mathcal{S}^H$ is constructed as follows: an edge of extremities n and m is introduced if, and only if, $\tilde{\Omega}_n$ and $\tilde{\Omega}_m$ intersects. The coarse graph corresponding to the partition of Fig. 2(a) is represented in Fig. 2(b).

Each subdomain $\tilde{\Omega}_n$ is extended to all the nearest nodes in order to define the domain Ω_n involved in the definition of the support of the coarse nodal function ϕ_n^H. Without such an extension, no degree of freedom would be available for the minimization problem and our method would coincide with the Reitzinger and Schöberl method (RS method).

For the 3-D eddy current problem, the formulation and the coefficients used are analogous to the 2-D case; the only difference is the chosen domain: the unit cube.

B. Results

The critical part of the computation is the construction of the coarse basis from the initial mesh. Results with decreasing $h_{\max}$ are given in Table I for the 2-D case and Table II for 3-D case. The number of unknows for system (11) is comparable to the number of DOFs in the 2-D case; it can be between twice and

TABLE I
RESOLUTION OF SYSTEM (11), 2-D CASE

	$h_{\max}$=0.2	0.1	0.05	0.025
Number of DOFs	119	401	1522	6143
Number of unknowns in (11)	84	226	1143	4670
CG iterations (A or S)	5	6	7	9
CG iterations (Id)	2	2	2	3

TABLE II
RESOLUTION OF SYSTEM (11), 3-D CASE

	$h_{\max}$=0.4	0.2	0.1	0.05
Number of DOFs	302	1559	11037	84403
Number of unknowns (11)	447	2910	25540	209087
CG iterations (A)	4	8	13	26
CG iterations (S)	5	11	13	22
CG iterations (Id)	3	5	5	6

TABLE III
MEAN NUMBER OF ITERATIONS AFTER 20 TIME STEPS, 2-D CASE

solver	$h_{\max}$=0.2	0.1	0.05	0.025
SSOR	26	36	67	135
RS method	4 (2)	7 (2)	11.5 (3)	18.5 (4)
without min.	4 (2)	6 (2)	10 (3)	16.5 (3)
Id	4 (2)	6 (2)	10 (3)	13 (3)
S	4 (2)	6 (2)	9 (3)	11 (3)
A	4 (2)	6 (2)	9 (3)	11 (3)

TABLE IV
NUMBER OF ITERATIONS FOR THE FIRST TIME STEP, 3-D CASE

solver	$h_{\max}$=0.4	0.2	0.1	0.05
SSOR	50	63	113	173
RS method	3 (2)	5 (2)	8 (2)	14 (3)
without min.	3 (2)	6 (2)	8 (2)	14 (3)
Id	3 (2)	5 (2)	8 (2)	14 (3)
S	3 (2)	5 (2)	8 (2)	14 (3)
A	3 (2)	5 (2)	7 (2)	13 (3)

three times the number of DOFs in the 3-D case. For the matrices K_e, we test several choices:

- matrices extracted from the global matrix A of the problem denoted by A in Tables I–IV;
- matrices extracted from the matrix defined from $\int_\Omega \delta \operatorname{curl} E \cdot \operatorname{curl} E'$ and a local regularization denoted by S in Tables I–IV;
- matrices are all equal to the identity, i.e., D is also the identity in (11), denoted by Id.

The number of nonpreconditioned CG iterations needed to divide by 10 the norm of the residual, when computing the minimization system (11), is almost independent of the mesh size in the 2-D case; in the 3-D case, it remains true for choice Id, but not for choice A.

The CG with various preconditioners is used for computing the solutions at each time:

- the classical SSOR method;
- the RS method;
- our multilevel method for the Id, A, and S cases and also for the case denoted "without min" where we only use the prolongation matrix built from the rows $(P_i^{\mathrm{edg}})'$.

For multilevel preconditioner, we use a V(1,1)-cycle [1] with the smoother proposed in [3]. The number of levels is given in brackets when applicable. The computation stops when the norm of the residual is divided by 10^{10}, the initial value is the solution for the previous time step. The mean number of iterations for this computation is favorable for our method as shown in Table III for the 2-D case. For the 3-D case, we only give the number of iterations for the first step in Table IV; the interest of our method is less obvious in this case, and the cost of solving (11) is not justified.

Observe that our method requires extra work to obtain the coarse basis compared to the RS method. Nevertheless, for time-domain computation, this initial effort is justified in the 2-D example, but not in the 3-D example.

VII. CONCLUSION

We propose an algebraic multilevel method for linear systems coming from incomplete first-order edge element discretization. Many parameters in this generic presentation can be tuned in view of specific applications. For instance, for the time-harmonic problem with $|\alpha h_{\max}^2| \ll 1$, the method denoted by S or by Id can be used. In order to balance the computational work between the construction of the coarse basis and the resolution of the initial system, the number of unknowns in problem (11) can also be decreased by removing some columns in B. Nonetheless, constraint (7) and, therefore, compatibility relation (3) is always ensured. Finally, an efficient implementation has to be implemented in order to test the methods on realistic examples.

REFERENCES

[1] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid.* San Diego, CA: Academic, 2001. with contributions by A. Brandt, P. Oswald and K. St'uben.
[2] R. Hiptmair, "Multigrid method for Maxwell's equations," *SIAM J. Numer. Anal.*, vol. 36, pp. 204–225, 1999.
[3] D. N. Arnold, R. S. Falk, and R. Winther, "Multigrid in $\mathbb{H}(\mathrm{div})$ and $\mathbb{H}(\mathrm{curl})$," *Numer. Math.*, vol. 85, pp. 197–217, 2000.
[4] R. Hiptmair, "Finite elements in computational electromagnetism," *Acta Numer.*, vol. 11, pp. 237–339, 2002.
[5] S. Reitzinger and J. Schöberl, "An Algebraic multigrid method for finite element discretizations with edge elements," *Numer. Linear Algebra Appl.*, vol. 9, pp. 223–238, 2002.
[6] P. Bochev *et al.*, "An improved algebraic multigrid method for solving Maxwell's equations," *SIAM J. Sci. Comput.*, vol. 25, no. 2, pp. 623–642, 2003. electronic.
[7] P. Bochev *et al.*, "Toward an h-independent algebraic multigrid method for Maxwell's equations," *SIAM J. Sci. Comput.*, to be published.
[8] W. L. Wan, T. F. Chan, and B. Smith, "An energy-minimizing interpolation for robust multigrid methods," *SIAM J. Sci. Comput.*, vol. 21, no. 4, pp. 1632–1649, 1999/00. electronic.
[9] P. Vanek, J. Mandel, and M. Brezina, "Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems,", citeseer.ist.psu.edu/vanek96algebraic.html, vol. 1, Tech. Rep. UCD-CCM-036, 1995. [Online]. Available.
[10] F. Musy, L. Nicolas, and R. Perrussel, "Gradient-prolongation commutativity and graph theory," *C. R. Acad. Sci. Paris*, vol. 341, no. 11, pp. 707–712, 2005.

Manuscript received June 25, 2005; revised November 21, 2005 (e-mail: ronan.perrussel@ec-lyon.fr).